

УДК 519.68: 004.434
DOI: 10.15827/0236-235X.128.690-695

Дата подачи статьи: 08.04.19
2019. Т. 32. № 4. С. 690–695

Компонент для интеграции исполняемых программ на языке программирования Kotlin в интернет-ресурсы

А.В. Прендота^{1,2}, студент, prendota@mail.ru

П.В. Балакшин², к.т.н., доцент, pvalakshin@gmail.com

¹ ИнтеллиДжей Лабс, г. Санкт-Петербург, 197374, Россия

² Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики (Университет ИТМО), г. Санкт-Петербург, 197101, Россия

Статья представляет собой обзор задач, методов и средств как обучения, так и работы с языком программирования Kotlin. Данный язык был создан как альтернатива другим популярным языкам программирования, работающим на базе JVM (Java Virtual Machine), а также применяемым при написании различных Android- и iOS-приложений. В силу нивелирования неудобств конкурентных языков программирования с 2017 года Kotlin стал официальным инструментом для операционной системы Android.

Показана важность создания и поддержки онлайн-среды программирования в качестве полноценного обучающего ресурса. Использование онлайн-среды в обучающих курсах дает возможность создавать примеры программ для изучения стилистических особенностей языка в виде кода, исполняемого непосредственно в браузере. Такой подход позволяет привлечь новых пользователей, а также обеспечить разработку новых проектов даже при отсутствии полноценной среды разработки на вычислительной машине программиста-разработчика, в том числе за счет наличия подсветки синтаксиса и автодополнения кода.

Представлен краткий анализ существующих онлайн-сред, выделены их недостатки, связанные с проблемами интеграции на сторонние сайты, малым количеством наглядных примеров, а также отсутствием подсветки синтаксиса и автодополнения кода. Кроме того, в статье предоставлена информация о способах решения ряда проблем, связанных с написанием и исполнением кода на языке Kotlin. Рассмотрены использование библиотеки Kotlin Playground, которая преобразует HTML-блоки в определенные редакторы кода, что дает возможность исполнять созданные редакторы непосредственно в браузере, а также такие функциональности библиотеки Kotlin Playground, как исполнение и компиляция кода под различные платформы, возможности разметки, автодополнения кода и подсветки, создание и исполнение тестовых сценариев, работа с API. Данный функционал позволяет пользователю адаптировать каждый компонент онлайн-среды под свои нужды и осуществлять грамотную интеграцию полученной среды в интернет-ресурсы.

Представлены примеры работы функций библиотеки Kotlin Playground, ее синтаксис, правила установки и дальнейшей загрузки с помощью Node Package Manager, а также использования данной библиотеки. В заключении статьи приведена ссылка на документацию и обозначены распространенные российские и международные образовательные онлайн-платформы, на которых уже реализованы или реализуются обучающие курсы по Kotlin с использованием библиотеки Kotlin Playground.

Ключевые слова: Kotlin, браузер, JVM, онлайн-платформа, API, online-среда разработки, автодополнение кода, подсветка синтаксиса.

Статически типизированный язык программирования Kotlin активно разрабатывается компанией JetBrains. Он работает на базе виртуальной машины Java Virtual Machine, широко распространенной в области промышленного программирования, и также компилируется в JavaScript. Изначально Kotlin разрабатывался с целью повышения безопасности и лаконичности по сравнению с языком программирования Java и для упрощения структуры языка по сравнению с языком программирования Scala. Ряд исследований подтверждают правильность вы-

бранного решения [1–4]. Kotlin был представлен в 2011 году, а уже в 2017 году компания Google сообщила, что язык программирования Kotlin стал официальным инструментом для операционной системы Android [5]. Данные факты подтверждают его актуальность, а также объясняют причины активного развития и роста популярности среди разработчиков по всему миру.

Необходимо отметить, что одними из важных аспектов развития любого современного языка программирования являются различные

курсы по изучению языка и удобная среда разработки для закрепления практических навыков. Компания JetBrains предоставляет среду IntelliJ IDEA для разработки на языке Kotlin, а также плагины для образования, которые позволяют непосредственно в IDE проходить различные курсы. Однако лучшим решением является онлайн-платформа. Многие разработчики популярных языков программирования поддерживают ее, чтобы можно было компилировать программы, не имея у себя на вычислительной машине полноценную среду разработки. Данная особенность обеспечивает регулярный приток новых пользователей. Следует отметить, что при наличии онлайн-платформы для исполнения кода в браузере можно создавать интерактивную документацию по языку программирования, а также по различным его библиотекам и фреймворкам, что позволит повысить визуальное восприятие технической документации.

Обзор существующих решений

К настоящему времени накопилось достаточно много различных сторонних платформ для исполнения кода большого числа языков программирования [3]. В таблице отражены результаты сравнения различных компиляторов для популярных языков программирования в глобальной сети Интернет, а также положительные и отрицательные стороны каждого из существующих решений.

Исходя из таблицы можно сделать вывод, что многие онлайн-компиляторы не поддерживают автодополнение кода и их нельзя интегрировать на сторонние сайты. Хотя в настоящее время существует множество популярных библиотек для подсветки синтаксиса, таких как Codemirror [6], highlights [7], некоторые платформы до сих пор используют черно-белое представление программного кода без синтаксических выделений, что заметно затрудняет восприятие читателем или разработчиком.

Библиотека для интеграции

Kotlin Playground – библиотека для создания исполняемых фрагментов кода на языке программирования Kotlin [8]. Главная задача библиотеки состоит в преобразовании HTML-блоков в определенные редакторы кода, которые могут исполняться непосредственно в браузере.

В качестве онлайн-редактора кода взята библиотека Codemirror [6], которая предоставляет API (Application programming interface) для подсветки синтаксиса языка, определения языковых отступов, маркировки текста.

Чтобы использовать библиотеку, достаточно добавить скрипт-элемент на веб-страницу:

```
<script src="https://unpkg.com/kotlin-playground@1" data-selector=".kotlin-code"></script>
```

По атрибуту data-selector указывается определенный html-селектор: тэг, уникальный идентификатор или атрибут класса. Это значит, что все элементы, которые содержат данный атрибут класса или html-тэг, будут преобразованы в исполняемые фрагменты кода, например, html-тэг code с атрибутом kotlin-code:

```
<code class="kotlin-code">
class Contact(val id: Int, var
email: String)
fun main(args: Array<String>) {
val contact = Contact(1,
"email@email.com")
println(contact.id)
}
</code>
```

Библиотека преобразует все тэги, помеченные классовым атрибутом со значением kotlin-code, в редактируемые и исполняемые фрагменты. Содержание этих фрагментов будет внутренней текстовой составляющей определенного html-блока.

Данная особенность обеспечивает инициализацию сразу нескольких фрагментов кода на

Особенности существующих онлайн-компиляторов

Peculiarities of existing online compilers

Компилятор	Подсветка синтаксиса	Автодополнение кода	Поддержка различных версий
Scala [https://scastie.scala-lang.org/]	Да	Нет	Да
Go [https://play.golang.org/]	Нет	Нет	Нет
Groovy [https://groovyconsole.appspot.com/]	Нет	Нет	Нет

веб-странице, так как может быть несколько тэгов с одинаковым классовым атрибутом и одинаковых html-тэгов, что является важнейшим фактором при создании технической документации по языку, потому что она включает в себя большое количество различных фрагментов кода на веб-странице.

Загрузить библиотеку можно из пакетного менеджера NPM [9, 10]. Для этого необходимо:

- добавить зависимость в проект с помощью команды

```
$ npm install kotlin-playground -S
```

- установить компонент библиотеки после того, как загрузится веб-страница, например:

```
import playground from 'kotlin-playground';
```

```
document.addEventListener('DOMContentLoaded', () => {
    playground('.kotlin-code');
});
```

где `kotlin-code` – классовый атрибут, к содержанию которого применяется алгоритм инициализации компонента библиотеки.

Исполнение кода под платформы JVM, JS

Язык программирования Kotlin компилируется под платформой JVM, а также может компилироваться в язык JavaScript [3, 11]. Библиотека Kotlin Playground поддерживает компиляцию Kotlin-кода под обе эти платформы.

Для компилирования кода под любую поддерживаемую платформу формируется объект, представляющий собой JSON-структуру следующего вида:

```
{
  args: "",
  platform: "",
  compilerVersion: ""
  files: [ { name: "", text: ""} ]
}
```

На сервер отправляется объект для компиляции с описанием платформы (`platform`), под которой должен компилироваться код, аргументы командной строки (`args`), если таковые присутствуют, версия компилятора языка программирования Kotlin (`compilerVersion`), а также массив файлов для компиляции (`files`), где в каждом файле указывается его имя (`name`) и текстовый код (`text`) на языке Kotlin.

Платформой по умолчанию в разработанной библиотеке является JVM. Когда пользователь отправляет запрос на компиляцию под данной платформой, сервер возвращает результат выполнения кода: JSON-объект с результа-

том выполнения кода или структуру случившегося исключения, ошибок компиляции программы. Пример работы библиотеки на веб-странице под платформой JVM представлен на рисунке (см. <http://www.swsys.ru/uploaded/image/2019-4/2019-4-dop/9.jpg>).

Для исполнения кода под платформой JS пользователю необходимо поставить дополнительный атрибут `data-target-platform` на html-тэг, который будет преобразован в исполняемый редактор кода, в значение `js`. После этого сервер не будет исполнять полученный код, а скомпилирует его в код на языке программирования JavaScript и вернет пользователю.

Полученный JavaScript код библиотека исполнит в отдельном `iframe` непосредственно в браузере и отобразит результат в консоли компонента.

Важной особенностью данной платформы является то, что пользователь может добавлять различные JavaScript-библиотеки и использовать их совместно с Kotlin-кодом. Для этого он должен в специальном атрибуте `data-js-libs` перечислить через запятую необходимые ему JavaScript-библиотеки. Все библиотеки, указанные в этом атрибуте, попадут в `header` фрейма под html-тэгом `script`. Как только все библиотеки будут загружены в `iframe`, полученный с сервера JavaScript код исполнится. Данная особенность позволяет создавать интерактивную документацию под различные библиотеки для языка Kotlin.

Разработка исполнения тестовых сценариев

Библиотека Kotlin Playground поддерживает исполнение тестовых сценариев. Для этого нужно указать на html-блоке атрибут `data-target-platform` со значением `JUnit`. Данная особенность дает возможность интеграции библиотеки в различные образовательные ресурсы: например, для построения онлайн-курсов по языку программирования Kotlin, где пользователю необходимо решить задачу и проверить свое решение. Когда пользователь использует платформу `JUnit`, сервер находит в отправленных ему классах тестовые методы и исполняет их, возвращая на клиентскую сторону объект с описанием результатов исполнения полученных тестов. На базе данной особенности построена Kotlin Koans [12] – популярная онлайн-платформа для изучения базовых особенностей языка Kotlin [11]. Пример работы библиотеки на веб-странице под

платформой JUnit представлен на рисунке (см. <http://www.swsys.ru/uploaded/image/2019-4/2019-4-dop/10.jpg>).

Kotlin Playground API

Библиотека Kotlin Playground предоставляет API для того, чтобы пользователь мог адаптировать каждый компонент. Например, пользователю интересно получить информацию о текущем редакторе (его платформу, аргументы, версию компилятора и многое другое) либо уведомление о том, что все тесты, которые он отправил на проверку, прошли успешно. Для этого необходимо добавить дополнительный параметр при инициализации компонента и в этом объекте передать нужные пользователю функции, например, чтобы получить состояние всего редактора, нужно передать дополнительную функцию `getInstance(instance)`, которая в параметре передаст текущее состояние редактора. На практике это будет выглядеть следующим образом:

```
function onTestPassed() {
    console.log("Tests passed!");
}
function getInstance(instance) {
    console.log(instance);
}
const options = {
    onTestPassed: onTestPassed,
    getInstance: getInstance(instance)
};
playground('.selector', options)
```

В первый параметр при инициализации компонента передается имя селектора, а во второй – список опций, которые обработает библиотека. В данном случае это опции получения состояния редактора и функция, которая будет срабатывать каждый раз после успешного выполнения предоставленных тестов.

С помощью API Kotlin Playground можно исполнять код, не нажимая на кнопку запуска, можно получать и смотреть на скомпилированный JavaScript код либо ловить различные события, например, закрытие или открытие консоли. Полный список опций содержится в документации к библиотеке [13].

Заключение

В статье представлена библиотека для интеграции исполняемых фрагментов кода для языка программирования Kotlin. На основе разработанной библиотеки построены online-среда разработки `play.kotl.in`, официальная документация по языку программирования Kotlin (<http://kotlinlang.org/docs/reference/>) (см. <http://www.swsys.ru/uploaded/image/2019-4/2019-4-dop/11.jpg>), образовательные курсы, которые можно встретить на платформах Stepik, Coursera, обучающие материалы по языку (`play.kotl.in/examples`), блоги и многое другое. Таким образом, интеграция данного компонента улучшает восприятие чтения, а также выразительность примеров кода: это позволяет пользователям не только видеть фрагменты кода, но и исполнять его.

Литература

1. Жемеров Д.Б., Исакова С.С. Kotlin в действии; [пер. с англ. А.Н. Киселева]. М.: ДМК-Пресс, 2017. 402 с.
2. Mateus B.G., Martinez M. An empirical study on quality of Android applications written in Kotlin language. *Empir. Soft. Eng.*, 2018, 39 p. DOI: 10.1007/s10664-019-09727-4.
3. Belyakova J. Language Support for generic programming in object-oriented languages: Peculiarities, drawbacks, ways of improvement. *LNCS*, 2016, vol. 9889, pp. 1–15. DOI: 10.1007/978-3-319-45279-1_1.
4. Flauzino M., Verissimo J., Terra R., Cirillo E., Durelli V.H.S., Durelli R.S. Are you still smelling it? A comparative study between Java and Kotlin language. *Proc. ACM Conf. SBCARS*, 2018, pp. 23–32. DOI: 10.1145/3267183.3267186.
5. JetBrains: Kotlin on Android. URL: <https://blog.jetbrains.com/kotlin/2017/05/kotlin-on-android-now-official/> (дата обращения: 01.03.2019).
6. Codemirror. URL: <https://codemirror.net/> (дата обращения: 01.03.2019).
7. Highlightjs. URL: <https://highlightjs.org/> (дата обращения: 01.03.2019).
8. Прендота А. Kotlin Playground. URL: <https://habr.com/company/JetBrains/blog/354620/> (дата обращения: 01.03.2019), (рус.).
9. Node Package Manager. URL: <https://www.npmjs.com/> (дата обращения: 01.03.2019).
10. Belov R. Embedding Kotlin Playground. URL: <https://blog.jetbrains.com/kotlin/2018/04/embedding-kotlin-playground/> (дата обращения: 01.03.2019).

11. Флэнаган Д. JavaScript. Подробное руководство; [пер. с англ. А.Н. Киселева]. СПб: Символ-Плюс, 2012. 1080 с.

12. Kotlin Koans. URL: <https://play.kotlinlang.org/koans/overview> (дата обращения: 01.03.2019).

13. Kotlin Playground. URL: <https://github.com/JetBrains/kotlin-playground> (дата обращения: 01.03.2019).

Software & Systems
DOI: 10.15827/0236-235X.128.690-695

Received 08.04.19
2019, vol. 32, no. 4, pp. 690–695

Component in the Kotlin programming language for integrating executable programs into internet resources

A.V. Prendota^{1,2}, Student, prendota@mail.ru

P.V. Balakshin², Ph.D. (Engineering), Associate Professor, pvbalakshin@gmail.com

¹IntelliJ Labs, St. Petersburg, 197374, Russian Federation

²The National Research University of Information Technologies, Mechanics and Optics, St. Petersburg, 197101, Russian Federation

Abstract. The paper provides an overview of tasks, methods and tools for training and working with the Kotlin programming language. This language was created as an alternative to other popular programming languages based on JVM (Java Virtual Machine), and also to the languages that were used to write various Android and iOS applications. Due to the fact that Kotlin leveled inconveniences of competitive programming languages, it has become the official tool for the Android operating system since 2017.

The importance of creating and supporting an online programming environment as a full-fledged training resource is shown. Using the online environment in training courses makes it possible to create sample programs for studying stylistic features of the language in the form of code that is executed directly in the browser. Such approach allows ensuring the development of new projects even in the absence of a full-fledged development environment on the developer's computer. Syntax highlighting and code completion also attract new users.

The paper presents a brief analysis of existing online development platforms and highlights their shortcomings associated with integration problems with third-party sites, a small number of illustrative examples, and the lack of syntax highlighting and code completion. Furthermore the paper provides information on how to solve a number of issues associated with writing and executing code in the Kotlin language. The paper considers the use of the Kotlin Playground library, which converts HTML blocks into specific code editors. This makes it possible to execute the created editors directly in the browser. Kotlin Playground library features such as code execution and compilation for various platforms, markup, code completion and highlighting capabilities, test script creation and execution, and working with the API are also considered. These features allow users to adapt every component of the online environment to his/her needs and to correctly integrate the resulting environment into Internet resources.

The paper gives examples of the Kotlin Playground library functions, presents its syntax, installation and further loading rules using Node Package Manager, as well as the use of this library. The paper concludes with a link to documentation and identifies common Russian and international educational online platforms on which Kotlin training courses have been or are being implemented using the Kotlin Playground library.

Keywords: Kotlin, browser, JVM, online platform, API, online development environment, code completion, syntax highlighting.

References

1. Zhemerov D., Isakova S. *Kotlin in Action*. Manning Publ., USA, 2017. Russ. ed.: Moscow, DMK Press, 2017, 334 p.

2. Mateus B.G., Martinez M. An empirical study on quality of Android applications written in Kotlin language. *Empir. Soft. Eng.* 2018, 39 p. DOI: 10.1007/s10664-019-09727-4.

3. Belyakova J. Language Support for generic programming in object-oriented languages: Peculiarities, drawbacks, ways of improvement. *LNCS*. 2016, vol. 9889, pp. 1–15. DOI: 10.1007/978-3-319-45279-1_1.
4. Flauzino M., Verissimo J., Terra R., Cirillo E., Durelli V.H.S., Durelli R.S. Are you still smelling it? A comparative study between Java and Kotlin language. *Proc. ACM Conf. SBCARS*. 2018, pp. 23–32. DOI: 10.1145/3267183.3267186.
5. *JetBrains: Kotlin on Android*. Available at: <https://blog.jetbrains.com/kotlin/2017/05/kotlin-on-android-now-official> (accessed March 1, 2019).
6. *Codemirror*. Available at: <https://codemirror.net/> (accessed: March 1, 2019).
7. *Highlightjs*. Available at: <https://highlightjs.org/> (accessed: March 1, 2019).
8. Prendota A. *Kotlin Playground*. Available at: <https://habr.com/company/JetBrains/blog/354620/> (accessed March 1, 2019) (in Russ.).
9. *Node Package Manager*. Available at: <https://www.npmjs.com/> (accessed March 1, 2019).
10. Belov R. *Embedding Kotlin Playground*. Available at: <https://blog.jetbrains.com/kotlin/2018/04/embedding-kotlin-playground/> (accessed March 1, 2019).
11. Flanagan D. *JavaScript: The Definitive Guide*. O'Reilly Media, USA, 2011. Russ. ed.: St. Petersburg, Symbol-Plus Publ., 2012, 1080 p.
12. *Kotlin Koans*. Available at: <https://play.kotlinlang.org/koans/overview> (accessed March 1, 2019).
13. *Kotlin Playground*. Available at: <https://github.com/JetBrains/kotlin-playground> (accessed March 1, 2019).

Для цитирования

Прендота А.В., Балакшин П.В. Компонент для интеграции исполняемых программ на языке программирования Kotlin в интернет-ресурсы // Программные продукты и системы. 2019. Т. 32. № 4. С. 690–695. DOI: 10.15827/0236-235X.128.690-695.

For citation

Prendota A.V., Balakshin P.V. Component in the Kotlin programming language for integrating executable programs into internet resources. *Software & Systems*. 2019, vol. 32, no. 4, pp. 690–695 (in Russ.). DOI: 10.15827/0236-235X.128.690-695.